SPECIFICATION

Electronic Version 1.2.8

Stylesheet Version 1.0

SPAM DETECTOR DEFEATING SYSTEM

Cross Reference to Related Applications

This application claims the benefit of the U.S. Provisional Application No. 60/235,433, filed on September 26, 2000.

Background of Invention

[0001] Spam, or unwanted emails and web pages can cause problems, including lost productivity based on the time that a user spends reading the spam. It is often desired to remove or block these messages. Different systems attempt to do so.

[0002] For emails, certain filtering systems exist. These filtering systems often work on the address level; i.e. certain users are blocked from sending further emails. My co-pending application no. 09/690,002 also describes another system which uses rules to remove Spam.

[0003] Spam can take another form -- specifically unwanted web pages. Certain web pages cause other web pages to open as so-called pop up windows. The theory is that a user will look at these, at very least while closing the window. Certain pop up window detectors such as POW!, available from www.analogx.com, kills unwanted pop ups immediately when they occur. However, POW! operates by the same system as disclosed above: specifically it detects an address which is programmed into a database of addresses, and uses that to make the decision to close the primary window.

Summary of Invention

[0004] The present application teaches different ways of defeating such systems as well as different countermeasures, which might defeat the defeating systems.

Brief Description of Drawings

[0005] These and other aspects will now be described in detail with reference to the

accompanying drawings wherein:Figure 1 shows a client and server connected via the Internet;Figure 2 shows a spam pop-up;Figure 3 shows a spam email;Figure 4 shows a flowchart of sending spam;Figure 5 shows a first spam defeating system;Figure 6 shows a way of distinguishing spam.

Detailed Description

[0006] The basic structure is shown in Figure 1, which shows an Internet server 100, connected to the Internet 110. The Internet server runs a program which can include an Internet server program such as Apache or IIS, and/or an email server or communication program. The server can carry out operations which are known in the art to either open pop up windows, or send Spam (unsolicited) email, or other unrequested advertising actions to the client 120.

[0007] Figure 4 shows a first flowchart which is operated by a sender, to send "Spam"; where Spam can be any communication, e.g. an email, web page, or other electronic communication which automatically sent to a user, without being specifically requested by the user, and can especially include advertising-oriented communications of this type. Examples of Spam include unsolicited emails, emails sent from an email mailing list, and pop up Internet windows.

[0008] The described system attempts to defeat these conventional ways of detecting Spam emails. At 400, the system determines a set of random elements. These can be random numbers, random characters, or any other random element. This can be based on a random number generator, or a random seed. Any ASCII character can be used, or only numbers or letters or any subset thereof.

[0009] At 405, the random number is incorporated into the Spam in some way, and becomes part of the Spam message, as explained below.

[0010] Figure 2 shows a pop up window. In a first embodiment, the random number 200 is used as part of the web page name 199. Therefore, the web page name either is the random number itself, or incorporates the random number as part of the name. The content is shown as 205. Here it says, "this is a Spam pop up page". The content may also include the random character therein.

[0011] Rule-based Spam-killing systems, such as disclosed in my application described above, simply look for information that fits the characteristics of a previously defined rule. This system, in contrast, changes the way the Spam looks, virtually every time it makes the Spam.

Therefore, this system may allow the Spam messages to come through, even when a rule based system is attempting to block them.

[0012] Certain "list based" detecting programs are specifically looking for the specific information that has been identified as part of the Spam. For example, POW may look for a web page having a name on a list. If a web page is named "Buy this book", and that term is on the list, then POW kills all web pages that are named that. Since this system names all the pop up windows differently (using the random character that will not, in general, be the same), that same specific information will not be found. Hence, these SPAM detectors will not detect that specific information and will not remove the Spam. Moreover, since a random number is generated, and a different random number may be used each time, the name always changes; and the conventional lists are not capable of preventing this Spam from reaching its target.

[0013] Figure 3 shows an alternative when used for creating email. The return address includes a random character, e.g., a random number, therein. It can include only the random character or the random character along with other information; shown as 300. The subject may also include the random character shown as 305. The body can also have the random character therein, shown as 310. The present system may work on Spam based emails, also.

[0014] Another embodiment discloses a technique to defeat such a random character based system. Figure 5 shows a system in which rules are written to determine the content of Spam. Again, the Spam can be in any description of electronic communication, e.g. in a pop-up page or in an email. According to these rules, the content being monitored is parsed into "words" at 505. These words can be different groups of characters which have spaces between them, or can be defined some other way such as by using a dictionary to find real words or just chunks of characters which form words, phonemes or any other unit.

[0015] At 510, an 80 or 90% fit is determined.

[0016] Alternatively, an exact fit of a specified number of characters, e.g., 15 characters, is determined. This latter system may be more useful when very long random characters are used.

[0017] When such a fit between the words being searched and the words in the email is determined, the message is determined to be Spam at 515. When the fit is not determined, the message is determined not to be Spam, and the message is delivered at 520. By operating to detect some coincidence less than 100%, e.g., 80-90%, the addition of random characters may

not defeat the system from detecting this kind of Spam, even though it does not that exactly meet the description in the list.

[0018] Another technique of detecting this kind of "random spam"is shown in Figure 6. The message is parsed into words at 600. The system detects gibberish, i.e. a series of random characters. This can be done by parsing the content into words which are compared against a dictionary. When the word is not within the dictionary (which can be a limited kind of dictionary if desired), then the word is established to be gibberish, and hence ignored, at 610. When the word is in the dictionary, the word is compared with the rules and/or list.

[0019] Another embodiment describes a way of defeating this kind of system described in Figure 6. This technique uses real words as the elements that are randomly-selected. The words are from within a dictionary of words. In this way, instead of the random characters being completely random, they include real words from a dictionary, but those real words are concatented in a random way. Either one word, or a number of words from a dictionary of words can be used. The words are randomly selected, thereby making these words randomly selected elements. Each message is still different; since each will contain different random words. Even if gibberish words are ignored, the rule based and/or list based systems may still fail to detect Spam that is marked in this way.

[0020] Still, each time the pop up window is made and/or a new Spam email is sent, random content is contained within that new window. In that way, it becomes more difficult for automated detectors to remove the Spam.

[0021] Other modifications are possible. For example, the descriptors may be any descriptor that is associated with a message; which may include, not only addresses, but also metatags, style sheets, or any other kind of information that is associated with a message.